# Alarm Talker©

Announce text messages to a single or multiple sound card destinations

User Manual

Revision: 1.7.0.0

Date: June 19, 2013

Copyright © 2013 Integrated Process Automation and Control Technologies Incorporated

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without written permission from IPACT Inc.

Alarm Talker Copyright IPACT 2013

file: https://svn.ipact.com:8443/svn/IPACT_Software/trunk/Alarm%20Talker/Doc/Alarm Talker.doc

| **Document Revision History** | |
|---|---|
| **Date** | **Description** |
| 30-Mar-2009 | M. Cochran Original |
| 31-Mar-2009 | E. Lakia Simple additions |
| 03-Sep-2009 | M. Cochran Version 1.1 convert to VS2008 |
| 29-sep-2009 | E. Lakia Added some results of actual testing |
| 30-Sep-2009 | M. Cochran version 1.2. Change the sleep time to half second between messages. |
| 21-Oct-2009 | M. Cochran version 1.3. Catch errors during speech synthesis and log error message. Add documentation to installation. |
| 27-May-2010 | Build installation kit as a 32 bit application only. Don't try to run as a 64 bit application on 64 bit machines. |
| 16-Aug-2010 | M. Cochran 1.4. Delete temporary wave files after playing them. |
| 30-Jan-2011 | M. Cochran 1.5 Fix problem with test utility. File delete was preventing play of message in the spawned thread. Wait for speech thread to play and exit before deleting the temporary file. Also had to set the target back to x86, it had reverted to Any CPU for some reason. It needs to be a 32 bit application because it is calling 32 bit routines for sound card interface. |
| 8-Feb-2011 | M. Cochran 1.6 Make poll rates for MSMQ messages and sound card configurable. Default them to 100 ms. |
| 19-June-2013 | M. Cochran 1.7 One deployment is having problems with messages not being annunciated a few days after the service has been started. A restart of the service would clear the problem. I believe that some error is causing a queue thread to exit. Make sure all potential errors are trapped and logged. Also, prevent any threads from exiting. |

Table of Contents


**1 Introduction to the Alarm Talker Service ..... 4**

1.1 TYPICAL APPLICATIONS ..... 4

**2 Installation ..... 5**

**3 Configuration ..... 6**

3.1 CONFIGURING SOUNDCARDNAME ..... 9

**4 Creating a MSMQ queue ..... 9**

4.1 GRANT QUEUE PERMISSIONS TO SYSTEM ..... 9

4.2 GRANT QUEUE PERMISSIONS TO ADDITIONAL ACCOUNTS ..... 10

**5 Starting ..... 10**

**6 Logging ..... 12**

**7 TTS Voice Engines ..... 14**

**8 Messages to Speak ..... 14**

**9 Alarm Talker Test Utility ..... 15**

**10 Sound Cards ..... 16**

# 1 Introduction to the Alarm Talker Service

**Alarm Talker** provides a solution for annunciating text messages via a text to speech engine. It supports multiple sound cards and accepts input messages from Microsoft Message Queue. Each queue can be directed to a different sound card. A single server may have multiple sound cards or USB attached sound devices.

Alarm Talker is a managed code windows service that was developed in Visual Studio 2005. It requires the .NET framework version 2.0. This framework is installed by the Alarm Talker installation setup program, if it has not previously been installed.

This product release has been validated on Windows XP and VISTA. Quality of sound devices can vary dramatically. However, if the device works with the Windows operating system, the device should work with Alarm Talker.

Alarm Talker is a 32 bit application. It will install and run as a 32 bit application on 64 bit machines.

## 1.1 Typical Applications

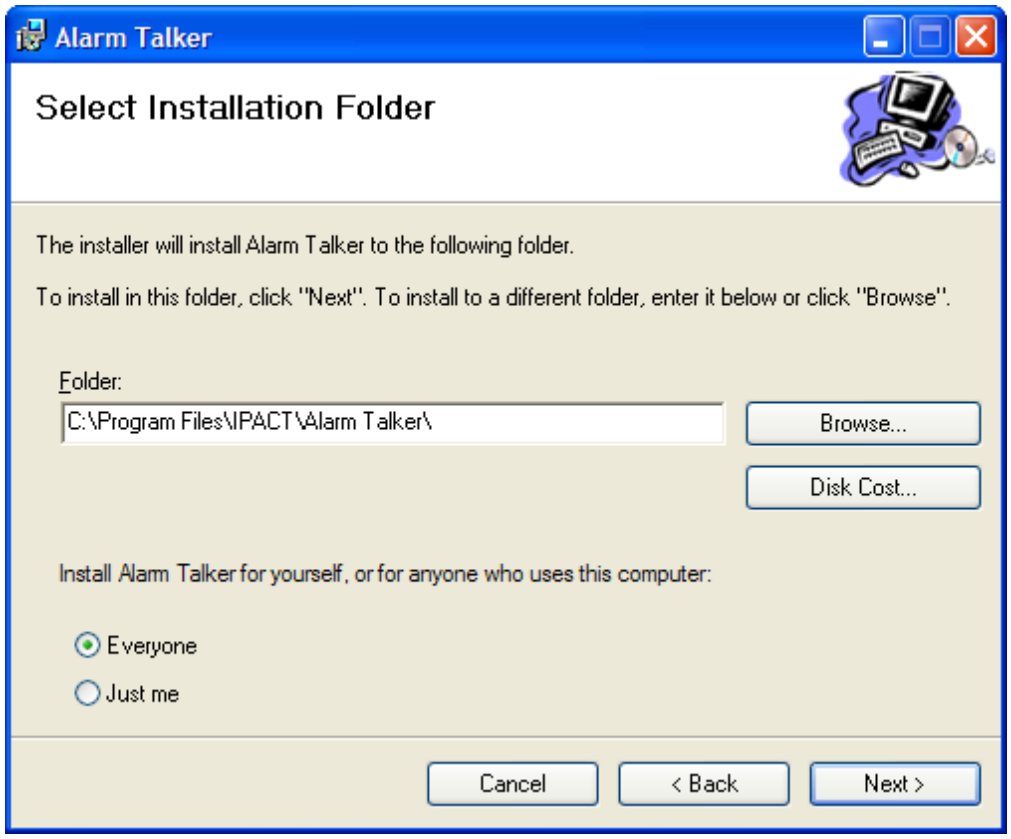
Alarm Talker can be used for announcing any text to public address systems, operators, or any place computer generated speech is required. A typical use is in the area of annunciation of process alarms to operators and others. Typically the sound card would be amplified from the output of the sound card. IPACT has also used this function from HP OpenVMS (uses IQRMSQ router to route IQR messages from OpenVMS http://www.ipact.com/Products/IQR.htm). SCADA systems such as iFIX that can write to MSMQ can also route text to this service.

An intermediate application could also be written that might merge data or play canned messages from a database and then queue the message on to be spoken.

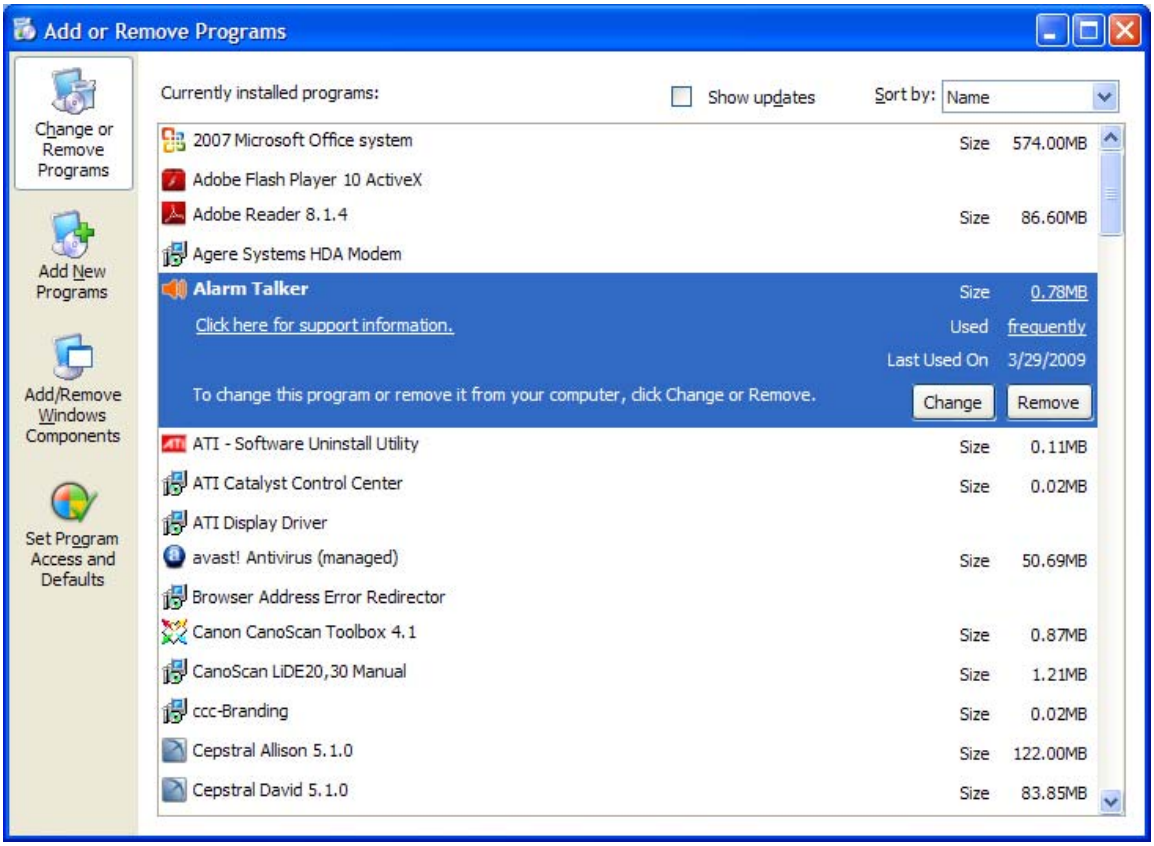
# 2 Installation

Prior to installing Alarm Talker, the system must have MSMQ installed. See:
http://msdn.microsoft.com/en-us/library/aa967729(VS.85).aspx

Alarm Talker installation is available as both a setup.exe and Alarm Talker Setup.msi files. The installation is simple. The following screen shot illustrates the primary decisions that you will need to make.


Alarm Talker
Select Installation Folder
The installer will install Alarm Talker to the following folder.
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".
Folder:
C:\Program Files\IPACT\Alarm Talker\
Browse...
Disk Cost...
Install Alarm Talker for yourself, or for anyone who uses this computer:
Everyone
Just me
Cancel
< Back
Next >

To remove the Alarm Talker use *Start → Settings → Control Panel → Add or Remove Programs* and select Alarm Talker as in the following screenshot. Select remove to remove the product.




# 3 Configuration

Alarm Talker is configured via an XML file located in the installation directory. A shortcut to this configuration is added to the *Start → Programs → Alarm Talker* folder during installation. You may edit this configuration file using any text editor of your choosing. An XML editer (i.e. notepad++) is recommended, but standard notepad is fine.

The following is an example configuration file.

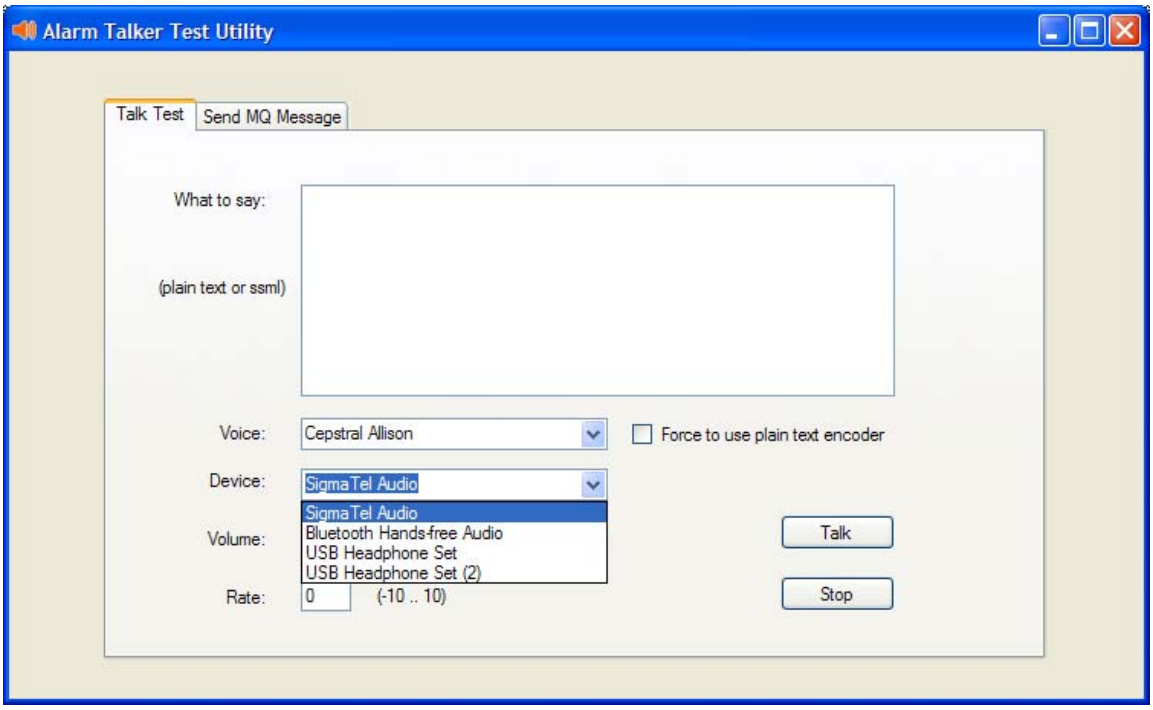
```
<AlarmTalkerConfig>
  <QueuePollRate>100</QueuePollRate>
  <SoundCardPollRate>100</SoundCardPollRate>
  <SoundCards>
    <SoundCard>
      <SoundCardId>1</SoundCardId>
      <SoundCardName>SigmaTel Audio</SoundCardName>
      <SoundCardDescription>Entry</SoundCardDescription>
      <Voice>Cepstral Allison</Voice>
      <Volume>100</Volume>
      <Rate>0</Rate>
      <ForcePlain>0</ForcePlain>
    </SoundCard>
    <SoundCard>
      <SoundCardId>2</SoundCardId>
      <SoundCardName>USB Headphone Set</SoundCardName>
      <SoundCardDescription>Exit</SoundCardDescription>
      <Voice>Cepstral David</Voice>
      <Volume>100</Volume>
      <Rate>0</Rate>
      <ForcePlain>0</ForcePlain>
```

```
    </SoundCard>
    <SoundCard>
      <SoundCardId>3</SoundCardId>
      <SoundCardName>USB Headphone Set (2)</SoundCardName>
      <SoundCardDescription>ASRS</SoundCardDescription>
      <Voice>Cepstral Allison</Voice>
      <Volume>100</Volume>
      <Rate>0</Rate>
      <ForcePlain>0</ForcePlain>
    </SoundCard>
  </SoundCards>
  <IncomingQueues>
    <IncomingQueue>
      <QueueName>.\private$\Alarm1</QueueName>
          <TargetSoundCards>
            <TargetSoundCard>
           <TargetSoundCardId>1</TargetSoundCardId>
            </TargetSoundCard>
      </TargetSoundCards>
    </IncomingQueue>
    <IncomingQueue>
      <QueueName>.\private$\Alarm2</QueueName>
          <TargetSoundCards>
            <TargetSoundCard>
           <TargetSoundCardId>2</TargetSoundCardId>
            </TargetSoundCard>
      </TargetSoundCards>
    </IncomingQueue>
    <IncomingQueue>
      <QueueName>.\private$\Alarm3</QueueName>
          <TargetSoundCards>
            <TargetSoundCard>
           <TargetSoundCardId>3</TargetSoundCardId>
            </TargetSoundCard>
      </TargetSoundCards>
    </IncomingQueue>
    <IncomingQueue>
      <QueueName>.\private$\Alarm4</QueueName>
          <TargetSoundCards>
            <TargetSoundCard>
           <TargetSoundCardId>1</TargetSoundCardId>
            </TargetSoundCard>
            <TargetSoundCard>
           <TargetSoundCardId>3</TargetSoundCardId>
            </TargetSoundCard>
      </TargetSoundCards>
    </IncomingQueue>
  </IncomingQueues>
</AlarmTalkerConfig>
<!--

 Alarm Talker configuration file.  This file is read during startup of the Alarm Talker.

 XML Header and Trailer

  The XML file must be wrapped with valid header and trailer tags to identify
  it as an Alarm Talker configuration file.

  <AlarmTalkerConfig>
  </AlarmTalkerConfig>

       QueuePollRate - How frequently each queue threads will peek for queue messages.
                       Tradeoff CPU usage vs timeliness of messages.
                       in milliseconds

       SoundCardPollRate - How Frequently sound card threads will check for messages to
                           annunciate.
                           Tradeoff CPU usage vs timeliness of messages.
                           in milliseconds
```

```
Sound card collection

<SoundCards>
  <SoundCard>
                see below...
  </SoundCard>
</SoundCards>


    SoundCardId - Any integer value to identify the card

    SoundCardName - The sound card as it is named by Windows.  See the names in the
                            drop down box on the Alarm Talker utility application.

    SoundCardDescription - Any text description you would like to give the card.  (i.e. pulpit, entry)

    Voice - The voice as is is named by Windows.  See the named in the drop down box on the Alarm
                        talker utility application.

    Volume - Integer value from 0 to 100.

    Rate - Integer value from -10 to 10.

    ForcePlain - Force to use the dot net SpeechSynthesizer.Speak instead of SpeechSynthesizer.Speakssml
                            method.  During development, found some TTS engines that render SSML text with the
                                                    plain .Speak method and not with the .SpeakSSML method.

<IncomingQueues>
  <IncomingQueue>
                see below...
  </IncomingQueue>
</IncomingQueues>

    QueueName - Name of the Microsoft Message Queue.

                <TargetSoundCards>
                  <TargetSoundCard>
                                see below...
                  </TargetSoundCard>
    </TargetSoundCards>

        TargetSoundCardId - Integer which refers to a SoundCardId defined in the SoundCards collection.

-->
```

## 3.1 Configuring SoundCardName

The SoundCardName field in the configuration file refers to the name that Windows assigns the sound card. This name can be retrieved from the Alarm Talker Test Utility. The following screen shot illustrates the sound card names that are installed. Any one of these names may be copied and pasted into the configuration file. The Test Utility may be used to interactivly determine which card is which by trial and error.



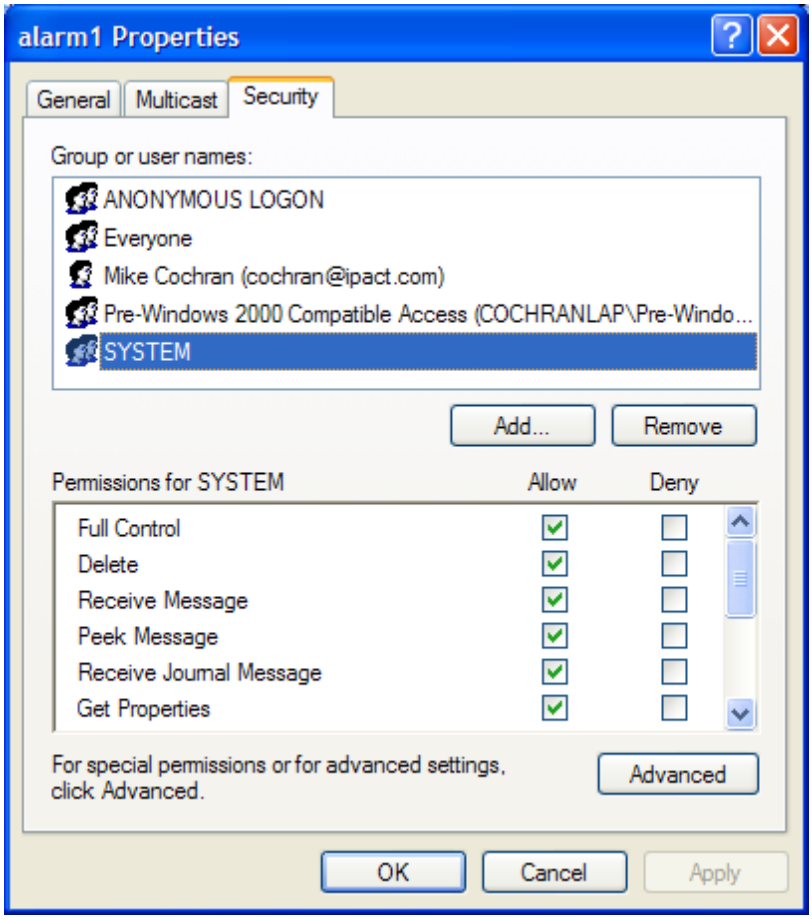

# 4 Creating a MSMQ queue

The input for Alarm Talker comes from MSMQ queues. Using the computer management utility found in: *Start → Control Panel → Administration Tools →Computer Management → Message Queuing → Private Queues, Right click on Private Queues* and select *New Private Queue. Enter any queue name and press OK (e.g., itn_to_clap).*

## 4.1 Grant queue permissions to SYSTEM

By default, the installation installs the Alarm Talker to run as “Local System”, which the local account is named “SYSTEM” on the install machine. The system account must have appropriate access to the MSMQ message queues.

To grant permission to access a queue, use computer management again. Right click on the queue (e.g., itn_to_clap) that you just created and choose properties and go to the security tab. Add SYSTEM to the list of users and give SYSTEM the desired permissions. The permissions should be granted to system for at least Peek, Receive and Send Message.

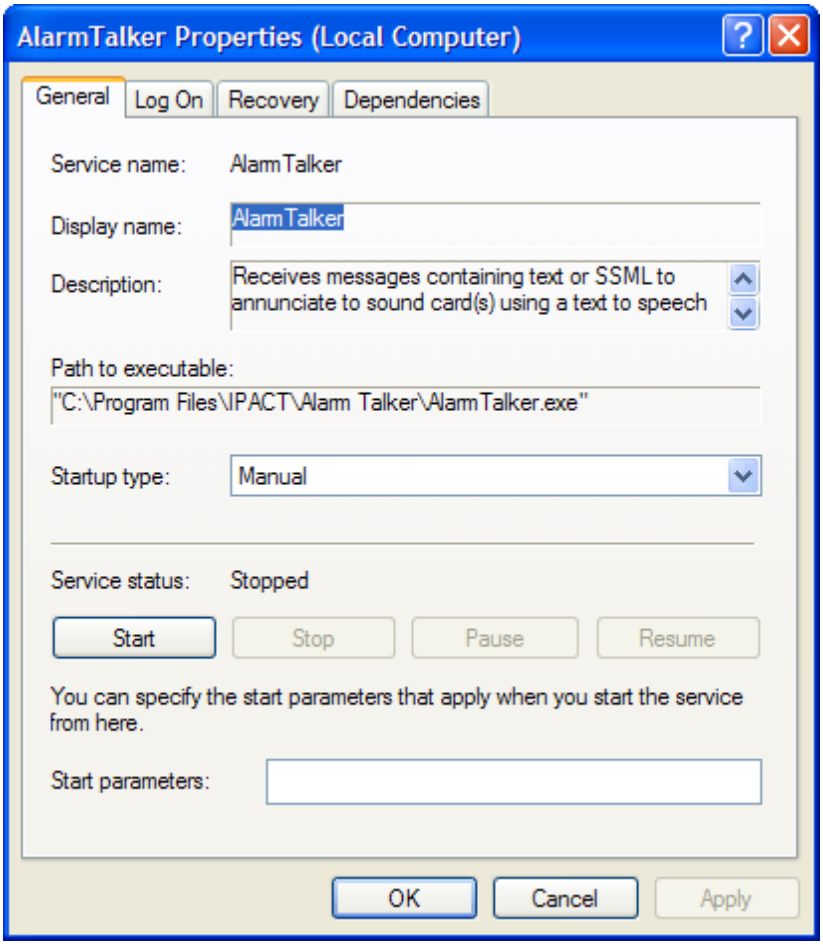



## 4.2 Grant queue permissions to additional accounts

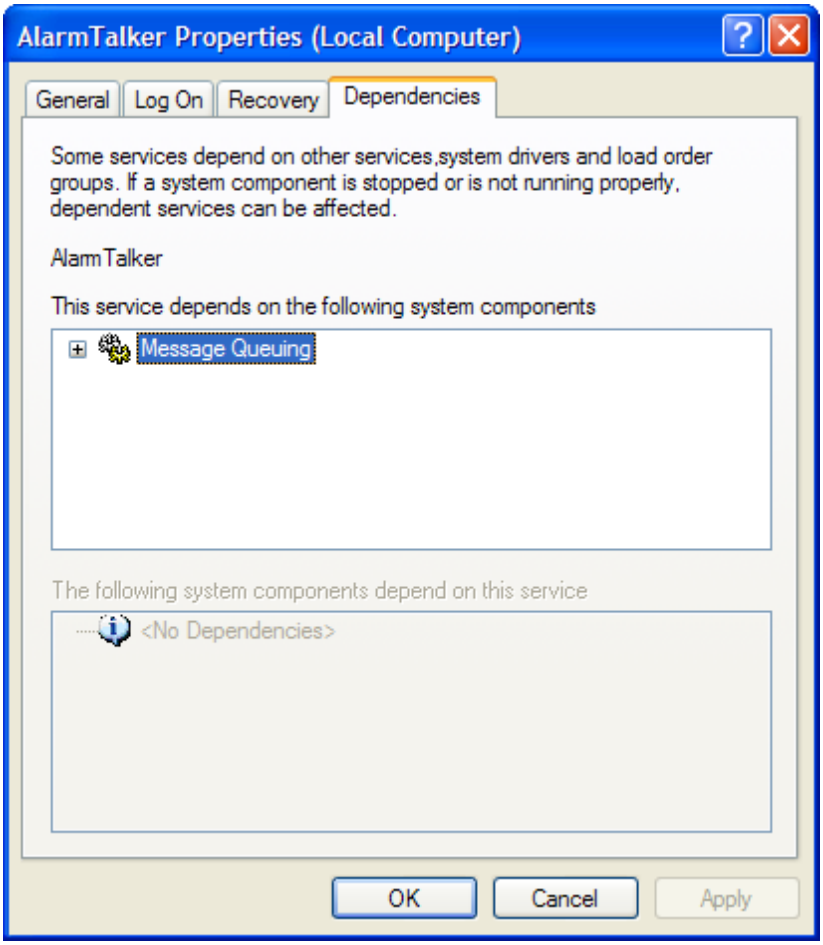
Any account that you would like to access the queues should be granted permissions to do so. These permissions should be granted to the account for at least Peek, Receive and Send Message,

# 5 Starting

The Alarm Talker service is started by using the Service Control Manager (SCM). It can be found in *Start → Control Panel → Administration Tools → Services*. It is listed as *Alarm Talker.* The following screenshot displays the service as it should appear after installation. You may change the startup type to automatic should you desire to start it automatically at the time that the computer boots.

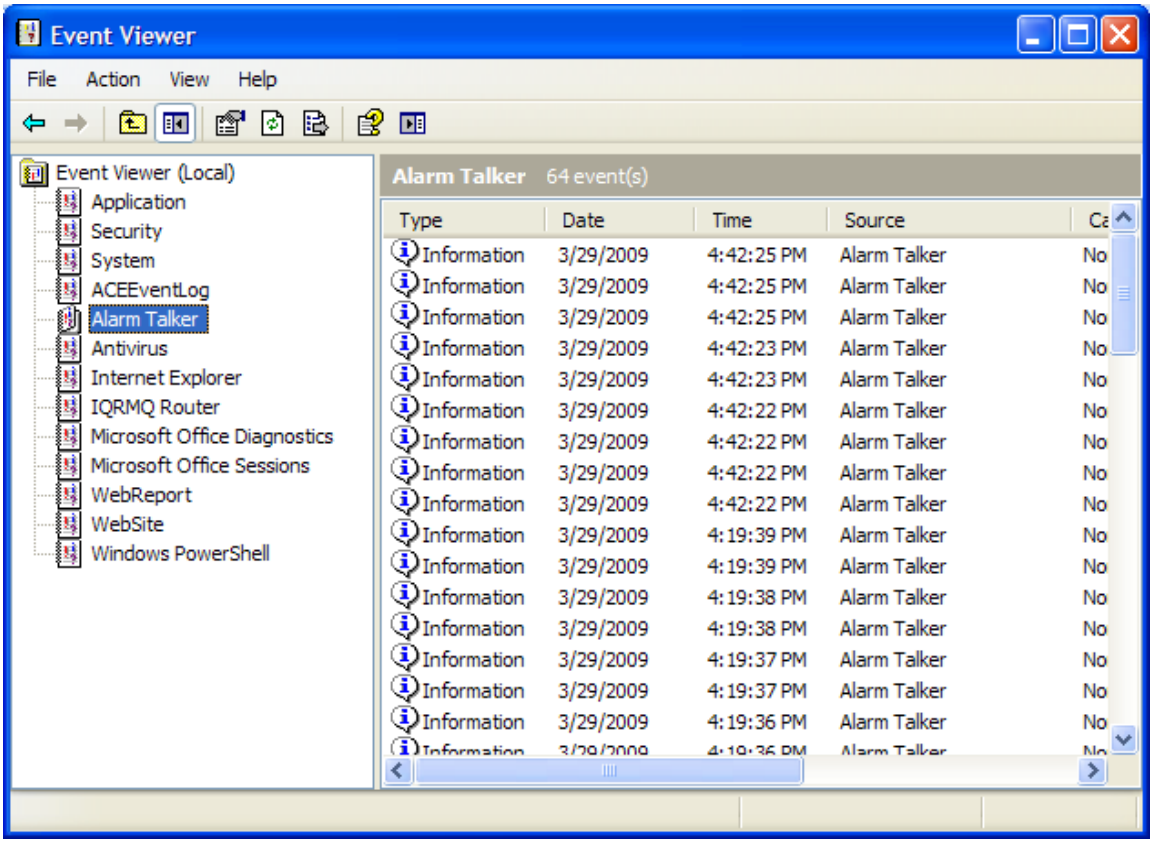



Note that Alarm Talker is dependent upon the Message Queuing service. When you execute start Alarm Talker Service, the service control manager will automatically start the Message Queuing service if it is not running. If starting Message Queuing fails, the Alarm Talker service cannot be started.

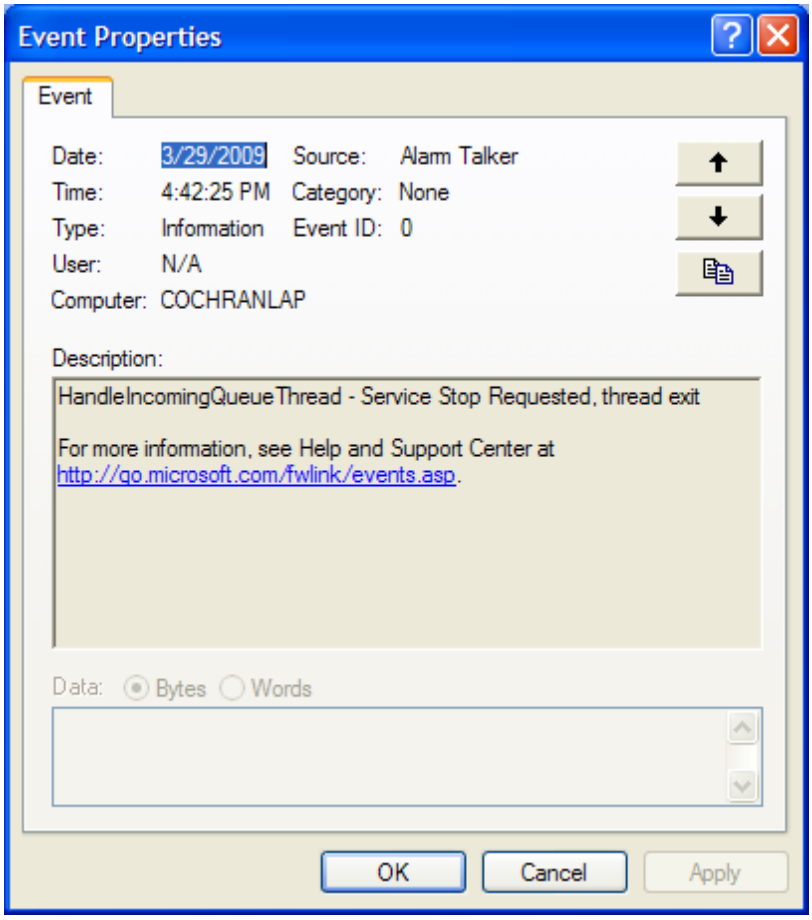

AlarmTalker Properties (Local Computer)
General
Log On
Recovery
Dependencies
Some services depend on other services,system drivers and load order groups. If a system component is stopped or is not running properly, dependent services can be affected.
Alarm Talker
This service depends on the following system components
Message Queuing
The following system components depend on this service
<No Dependencies>
OK
Cancel
Apply

# 6 Logging

Alarm Talker prints log messages to the event logger. The custom log ***Alarm Talker*** is created the first time that it is run. The following screenshot illustrates this log.




Also note that the Windows Service Control Manager (SCM) will log messages to the ***Application*** log. On start, stop of the service. Also, if any catastrophic errors occur that cause the Alarm Talker to crash, they would be reported in the ***Application*** log.




# 7 TTS Voice Engines

Microsoft provides voice engines with their operating systems. As of this time, third party voice engines are superior to those engines. Microsoft maintains a list of the third party engines at this link:

http://www.microsoft.com/speech/evaluation/thirdparty/engines.mspx

Microsoft provides a Text To Speech engine with Windows XP and later. IPACT has tested the Cepstral product successfully.

# 8 Messages to Speak

The message to speak can be expressed as plain text or in Speech Synthesis Markup Language (SSML). Alarm Talker looks for the first character of the string to start with an angle bracket (<) for SSML, otherwise it assumes the message is plain text. SSML is treated differently during encoding the text. Specifically the `SpeechSynthesizer.SpeakSSML` method is used for SSML, unless for ForcePlain configuration setting is in effect. For plain text the `SpeechSynthesizer.Speak` method is used.

Here is an example of the SSML:

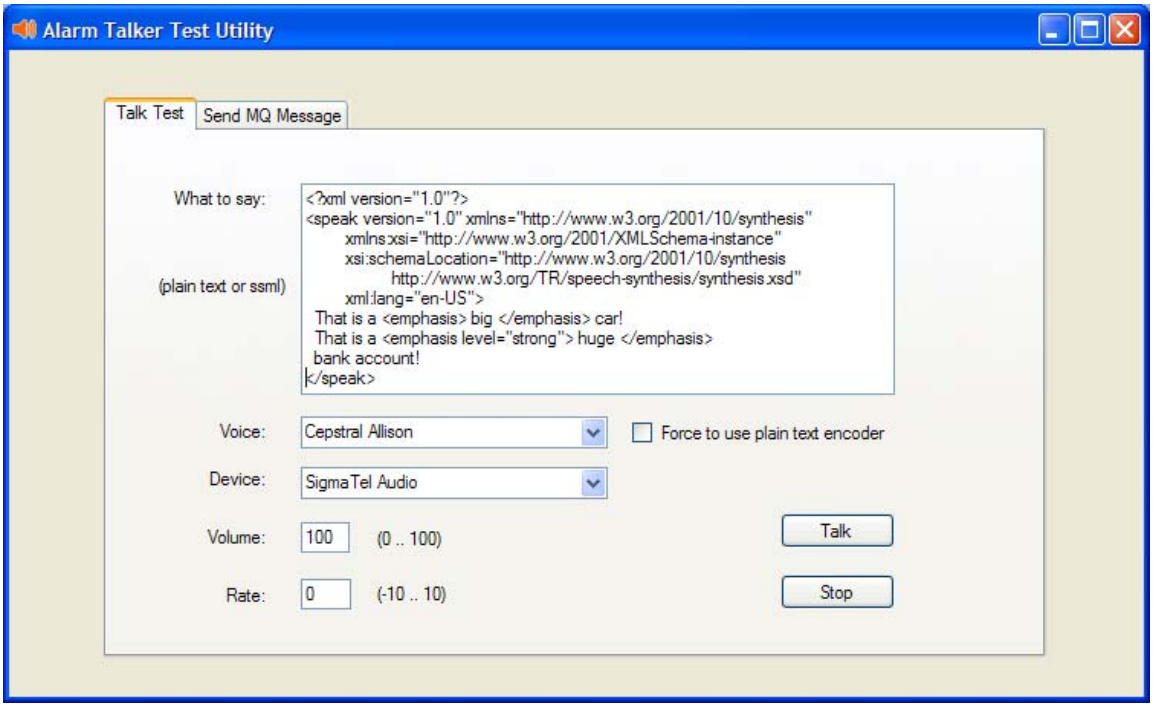
```
<?xml version="1.0"?>
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
                   http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
         xml:lang="en-US">
  That is a <emphasis> big </emphasis> car!
  That is a <emphasis level="strong"> huge </emphasis>
  bank account!
</speak>
```

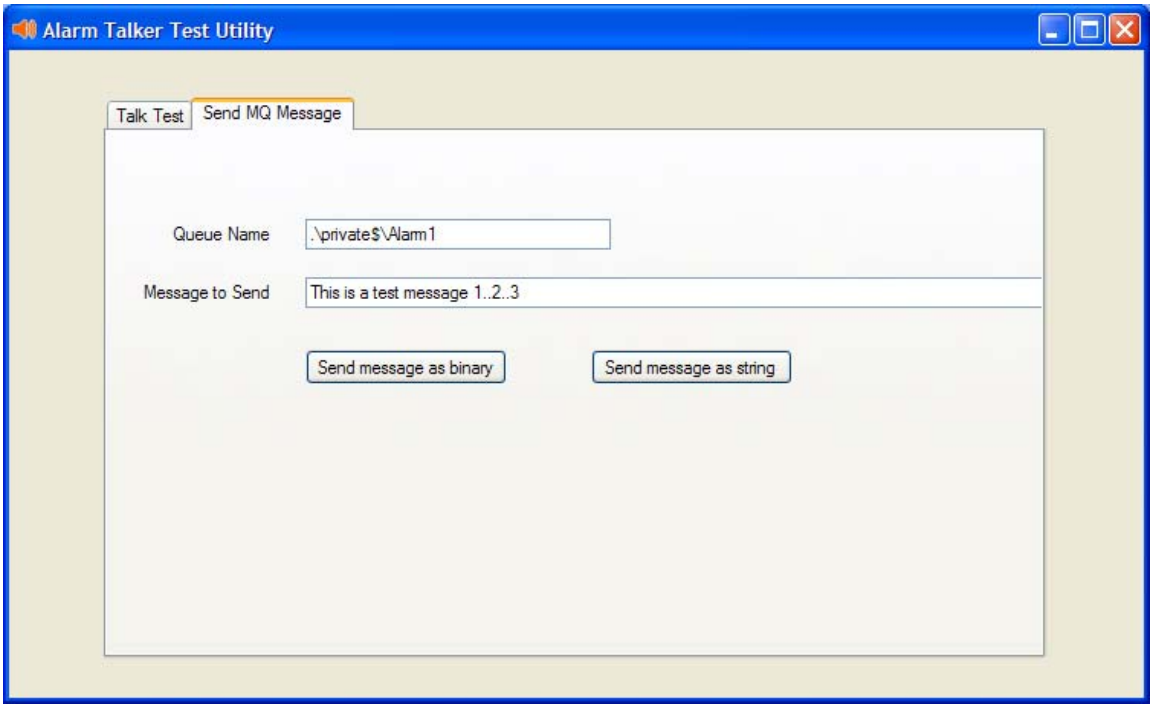
For more information on SSML, here is the W3C reference:

http://www.w3.org/TR/speech-synthesis/

# 9 Alarm Talker Test Utility

The Alarm Talker Test Utility can be accessed from *Start → Programs → Alarm Talker → Alarm Talker Test Utility.* It can be used to experiment interactively with constructing voice messages in both plain text and SSML. It can also be used to send these messages via MSMQ. The following are some screen images of this utility.






## 10 Sound Cards

Below, we list a few sources for sound cards with multiple channels. These are high quality cards with up to ten output channels. We have not tested these cards. We see no reason why they would not work. But make sure you investigate the details for these cards. Documentation for one of the cards is also listed.

http://www.zzounds.com/item--MDOD1010LT

http://www.altoedge.com/soundcards/index.html#AUDSCM8

Documentation: http://cachepe.zzounds.com/media/DELTA1010LTr3-2d71ed3d86b5e5c15f4af7249afb73e7.pdf

We have tested this application using USB sound card devices. It is our experience during testing that the USB devices will boot to consistent names to cards providing that the devices are not moved and no additional USB devices are attached to the USB. IPACT successfully tested the Delta 1010LT. The Delta 1010LT has four stereo output ports. These four ports and the on board (if available) sound output can be used.